



Algoritmos II

Dicionários

Prof. Fernando Del Moro



Dicionários

Dicionário são mais um exemplo de estrutura de dados capaz de armazenar múltiplos valores em apenas uma (1) variável;

Apesar do enquadramento, dicionários trazem especificações e detalhes únicos e que não são encontrados em outras estruturas;

A principal é que dicionários não usam índices e sim chaves (*key*) para armazenar seus dados;



Dicionários

As chaves (*key*) são literais (nomes) utilizados para identificar os valores colocados em um dicionário;

Índices não são reconhecidos e sua utilização causa erros;



Dicionários (quando usar)

A utilização de dicionários é indicada quando o número de valores armazenados em uma estrutura for grande e heterogênea a ponto de causar confusão se utilizada com índices;

Apesar disso, seu uso requer alguns cuidados e atenção;

Criação de Dicionários

A criação de dicionários tem caracteres especiais para indicar sua utilização sendo eles as chaves "{", "}", desta forma:

() → **TUPLAS**

[] → **LISTAS/VETORES/MATRIZES**

{ } → **DICIONÁRIOS**

Criação de Dicionários

A criação de dicionários pode ser:

```
dicionario = dict()
```

```
dicionario = {'chave':valor, 'outrachave':outro valor}
```

Observe a sintaxe.... 'chave':valor





Operações

Diferente de listas, não é possível fazer "append", apenas atribuições:

```
dicionario['chave'] = valor desta chave
```



Operações

Existem algumas informações relevantes que podem ser obtidas através dos comandos abaixo e que permitem diversas manipulações.

`dicionario.keys()`

Retorna lista de chaves

`dicionario.values()`

Retorna lista de valores

`dicionario.items()`

Retorna lista de chaves/Valores



Operações

Estes métodos podem ser percorridos por “for”

```
for k in dicionario.keys() :
```

```
for v in dicionario.values():
```

```
for i in dicionario.items():
```



Operações

Como não existem índices em dicionários o “for” com *enumerate* não funciona, sendo necessário uma sintaxe diferente:

```
for k,v in dicionario.items():
```



Operações

É possível apagar um valor de um dicionário com o comando "del":

```
del (dicionario['chave'])
```



Operações

Diferente das tuplas que são imutáveis é possível alterar valores mesmo depois de criado:

```
dicionario['chave'] = 'Novo Valor'
```

É possível também adicionar novas chaves/valores mesmo depois de criados, bastando apenas atribuir um valor a uma nova chave

```
dicionario['novaChave'] = novo valor
```



Operações

Não é possível fazer o “fatiamento” de um dicionário;

Quando for necessário uma cópia (para adicionar em listas, por exemplo) é necessário um método próprio:

```
dicionario.copy()
```




Exercícios

- 1) Faça um programa que leia nome e média de um aluno. Guarde também a situação em um dicionário. Mostre a estrutura na tela.
- 2) Crie um programa onde 4 jogadores joguem um dado e tenham resultados aleatórios. Guarde esses resultados em um dicionário. Mostre a lista dos jogos na tela.



Exercícios

3) Crie um programa que gerencie o aproveitamento de um jogador de futebol. O programa vai ler o nome e quantas partidas ele jogou. Depois vai ler a quantidade de gols feitos em cada partida. No final tudo isso será guardado em um dicionário, incluindo o total de gols feitos durante o campeonato



Exercícios

4) Crie um programa que leia nome, idade e sexo de várias pessoas, guardando os dados de cada pessoa em um dicionário e todos os dicionários em uma lista. No final, mostre:

- Quantas pessoas foram cadastradas
- A média de idade do grupo
- Uma listagem com todas as mulheres
- Uma listagem com todas as pessoas acima da média